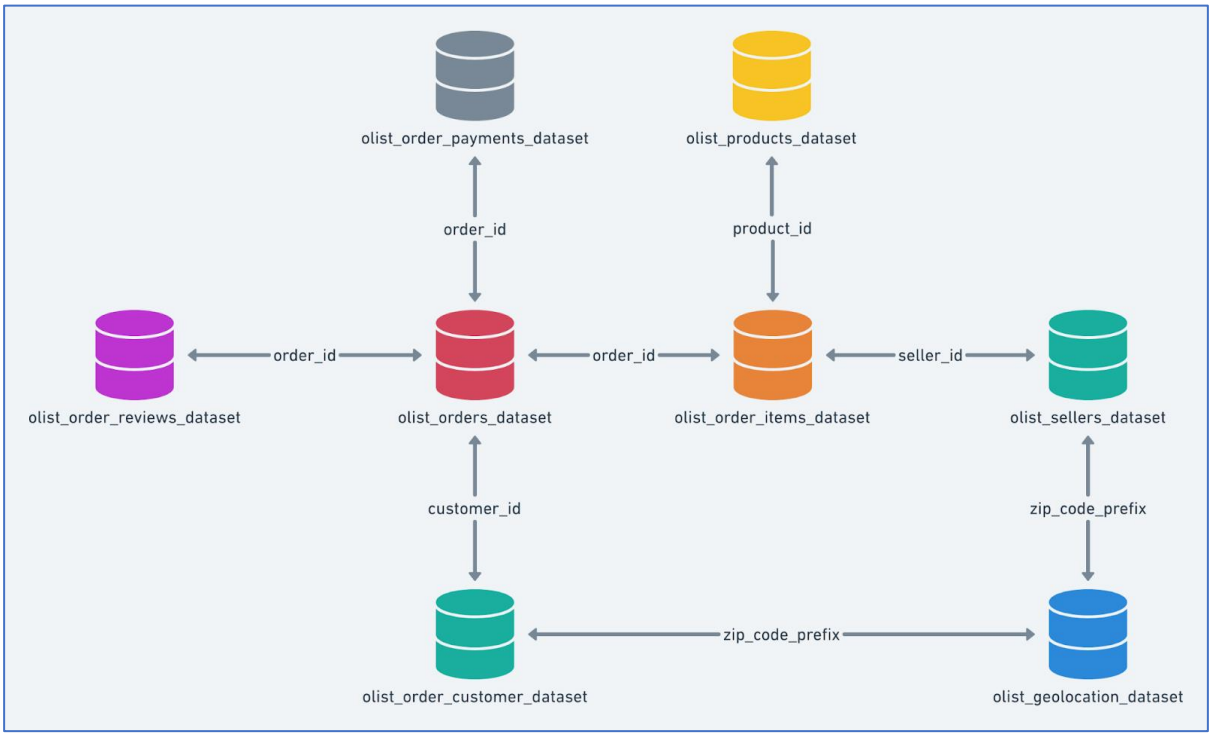
Business Case: Target SQL

Context:

Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

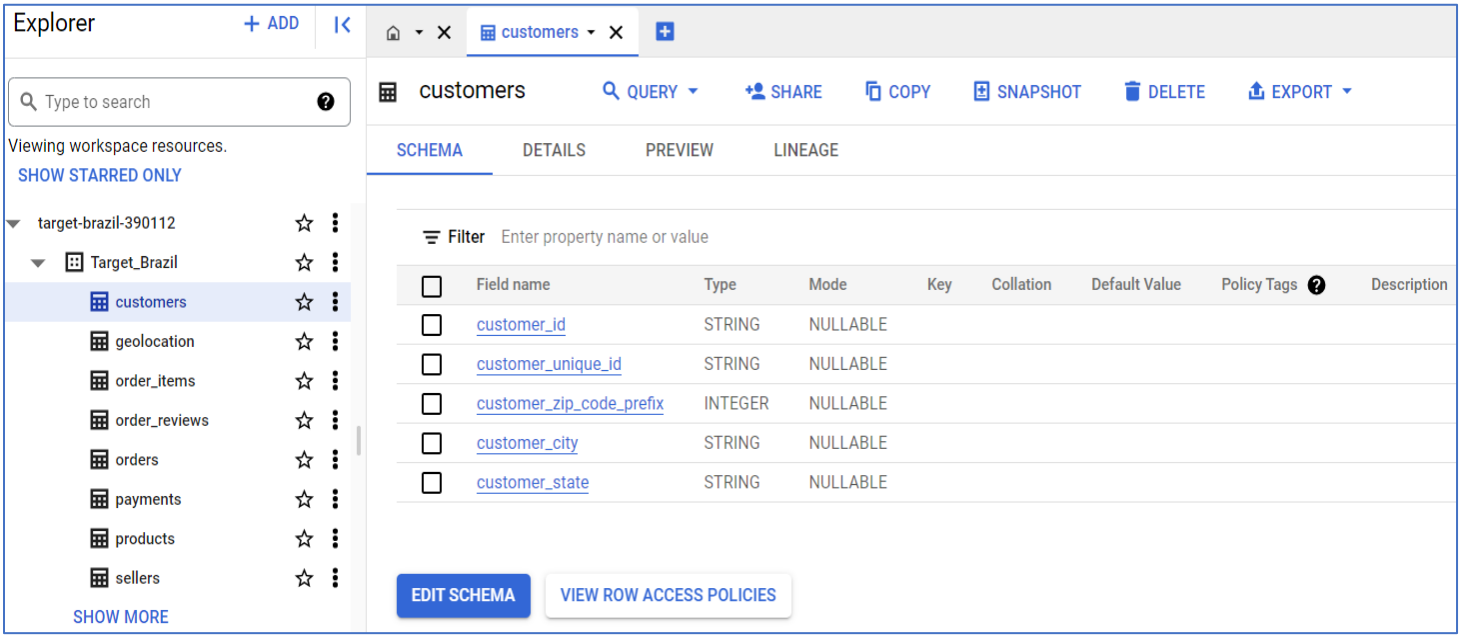


Note -

1. Project is named as Target_Brazil.
2. To create the dataset, the 8 CSV files provided for the Target Case Study were uploaded, and each table was named after its corresponding file name.

Q.1 Exploratory Analysis of the Dataset

A) Data type of all columns in the "**customers**" table



Steps

1. Click on Target_Brazil.
2. Then Click on Customers table.
3. Afterward, click on the "Schema" option
4. To identify the data type for the "Customers" table, refer to the provided screenshot and check the data present in the "Type" column

Answer

The Data type of all columns in the customer table is as follow – **String**, **Integer**

B) Time Range between which the orders were placed

Query - SELECT

MIN(order_purchase_timestamp) as Start_Date,

MAX(order_purchase_timestamp) as End_Date

FROM

`Target_Brazil.orders`

```
SELECT
  MIN(order_purchase_timestamp) as Start_Date,
  MAX(order_purchase_timestamp) as End_Date
FROM
  `Target_Brazil.orders`
```

Output/Query Result

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	Start_Date	End_Date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Steps

1. Click on the "**+**" symbol or the "**Compose New Query**" button.
2. A new tab will open where you can write your query.
3. To determine the time range during which the orders were placed, we will use the **MIN** and **MAX** functions from the "orders" table. These functions will help us identify the earliest and latest dates of orders.

Insights

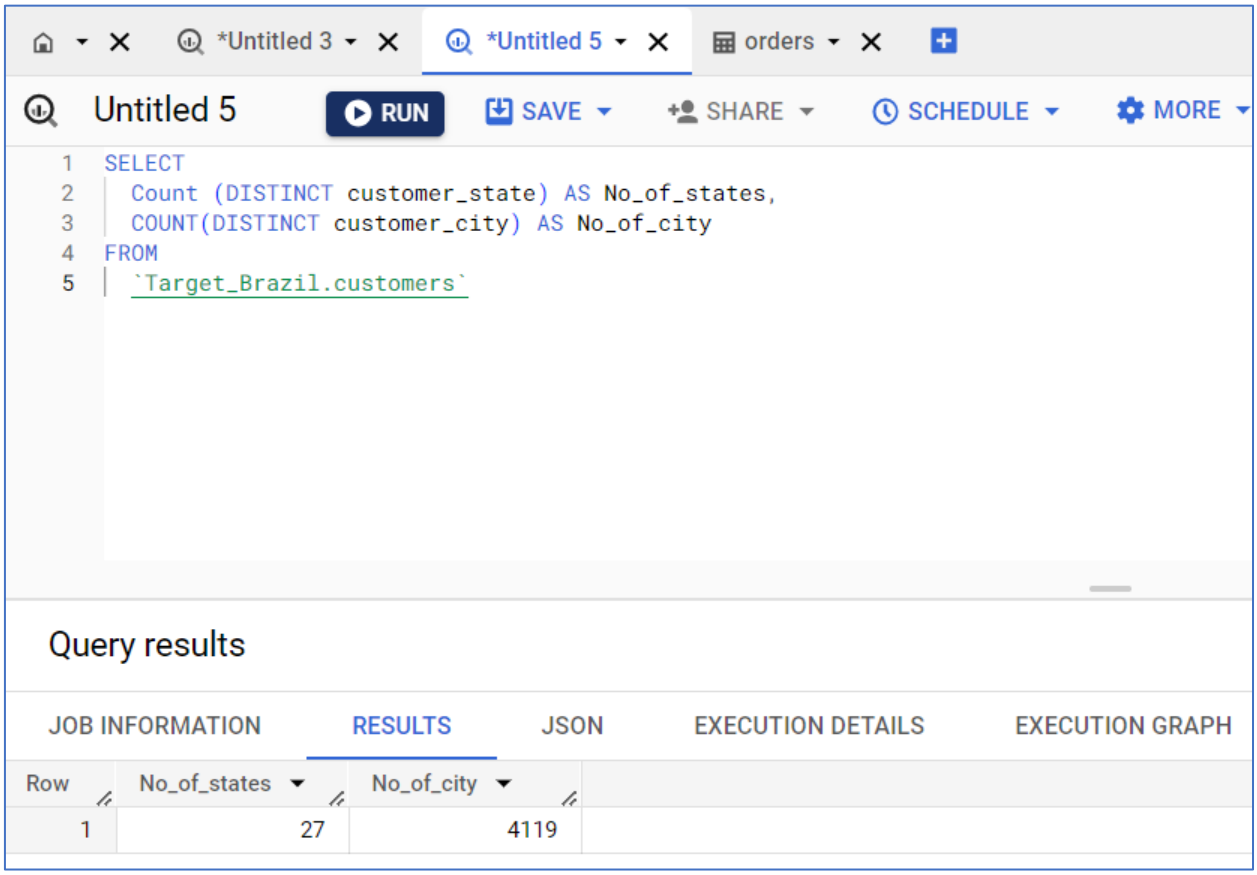
1. The Time Range between which the orders were placed are from **2016-09-04** to **2019-10-17**

2. The time range mentioned indicates that we have orders data starting from **September 2016** until **October 2019**, covering the period from the **Q3 2016** to the **Q4 2019**
3. The orders data is **only available for specific time periods**. Specifically, it includes information for quarters Q3 and Q4 of 2016, as well as all months of 2017, 2018, and the first ten months of 2019. However, there is **no data** for the first two quarters (Q1 and Q2) of 2016 and the last two months of Q4 in 2019

C) Count the number of Cities and States.

Query - SELECT

```
  Count (DISTINCT customer_state) AS No_of_states,
  COUNT(DISTINCT customer_city) AS No_of_city
FROM
  `Target_Brazil.customers`
```



Insights

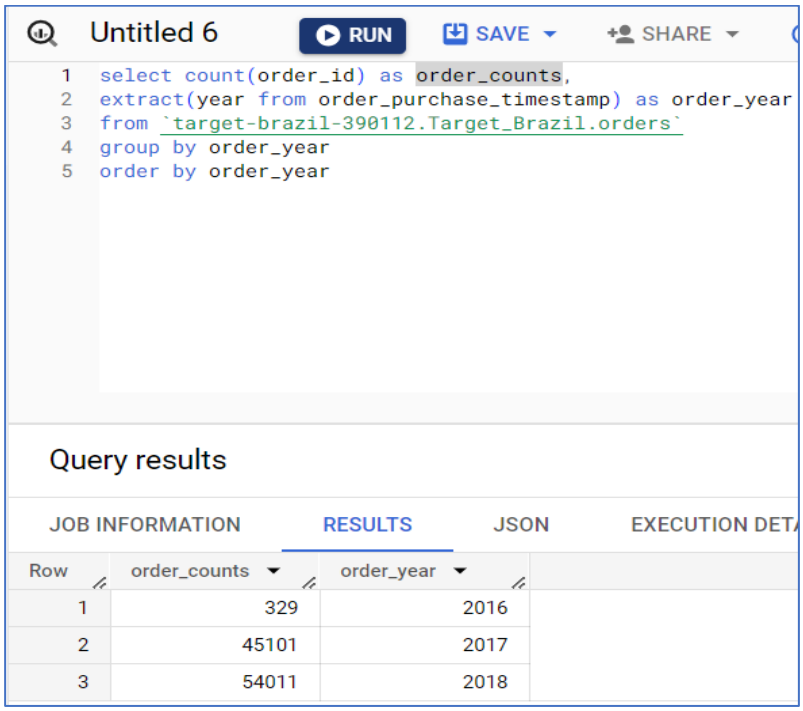
There are a total of **27** states and **4119** cities in our dataset

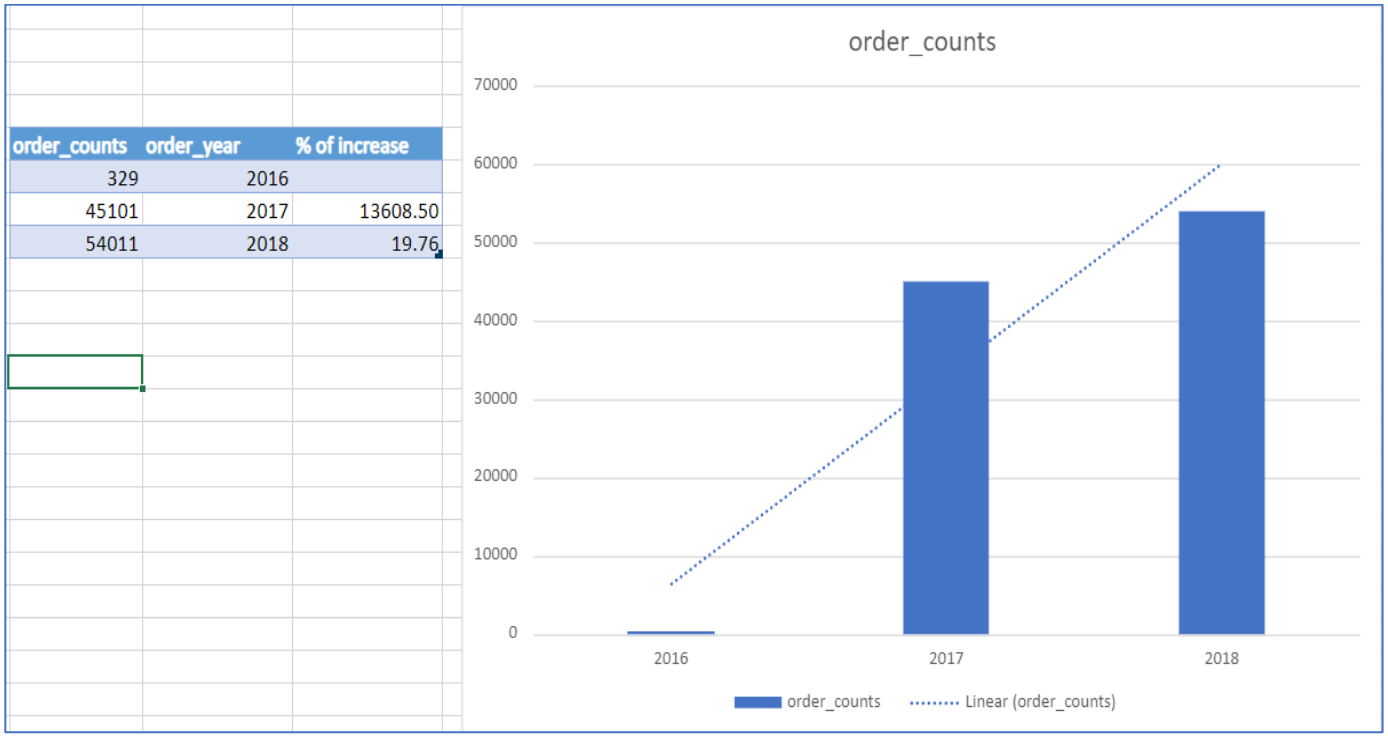
Q.2 In-depth Exploration of Dataset

A) Is there a growing trend in the no. of orders placed over the past years?

Query -

```
select count(order_id) as order_counts,
extract(year from order_purchase_timestamp) as order_year
from `target-brazil-390112.Target_Brazil.orders`
group by order_year
order by order_year
```





Insights

Yes, there is a growing trend in the number of orders placed over the past years.

1. In 2016, there were **329** orders.
2. In 2017, the number of orders increased to **45101**, representing a **13,608.50%** increase compared to the previous year.
3. In 2018, the number of orders further increased to **54011**, indicating a **19.76%** increase compared to the previous year.
4. These significant **year-on-year** increases strongly suggest a growing trend in the number of orders placed over the past years.

B) Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Query -

```
SELECT
  COUNT(order_id) AS order_counts,
  EXTRACT(year
  FROM
    order_purchase_timestamp) AS order_year,
  EXTRACT (month
  FROM
    order_purchase_timestamp) AS order_month
FROM
  `target-brazil-390112.Target_Brazil.orders`
GROUP BY
  order_year,order_month
ORDER BY order_counts desc,order_year,order_month
```

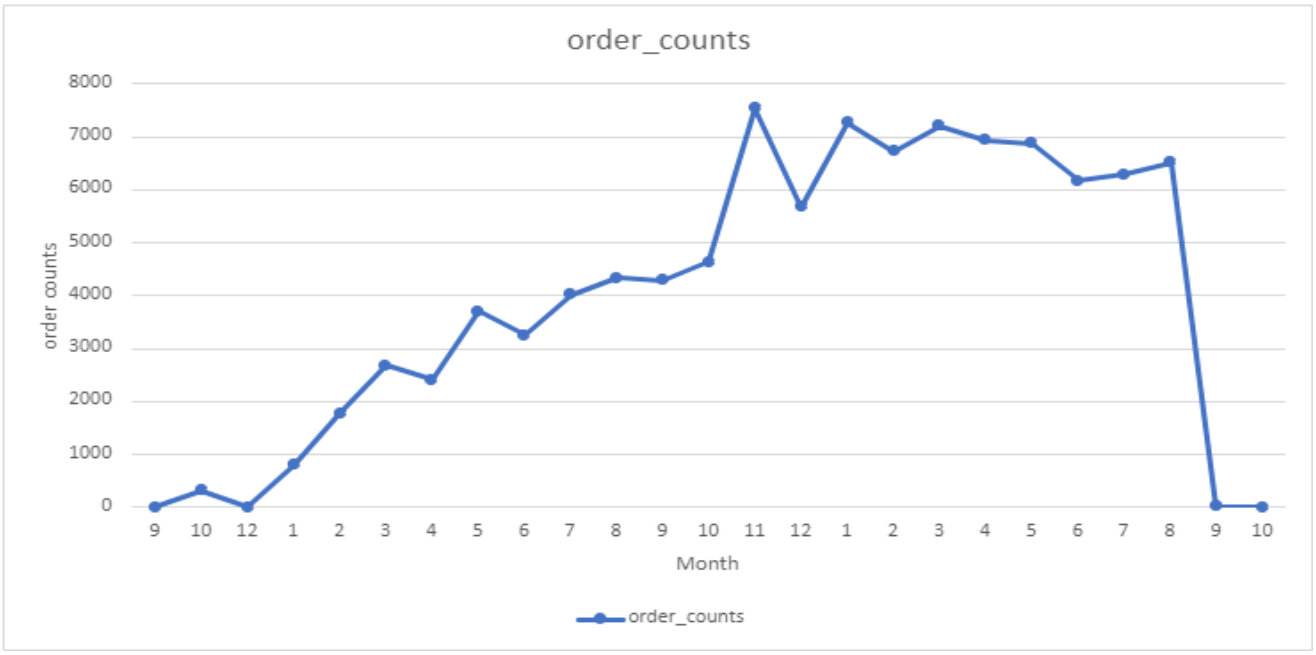
Untitled 7 RUN SAVE SHARE SCHEDULE MORE

```sql
SELECT
  COUNT(order_id) AS order_counts,
  EXTRACT(year
  FROM
    order_purchase_timestamp) AS order_year,
  EXTRACT (month
  FROM
    order_purchase_timestamp) AS order_month
FROM
  `target-brazil-390112.Target_Brazil.orders`
GROUP BY
  order_year,order_month
ORDER BY order_counts desc,order_year,order_month
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	order_counts	order_year	order_month
1	7544	2017	11
2	7269	2018	1
3	7211	2018	3
4	6939	2018	4
5	6873	2018	5
6	6728	2018	2



Insights

The chart displays a **cyclical pattern** in order placement, with alternating increases and decreases in the number of orders. Notably, November shows the highest **peak** in order volume, followed by a **decline** in December. The trend continues with a subsequent increase in January and continues in a similar fashion. However, in September and October of 2018, there was a significant and sharp decline in order volume

Recommendation

By examining the provided chart, we have identified the months with the highest order volumes and discerned a **recurring pattern** in customer order placement. This analysis allows us to pinpoint the specific months when the most orders are placed. Subsequently, we can conduct **market research** and **user studies** to understand the reasons behind the increased order volume during those months and the subsequent drop in the following

months. It is worth investigating whether we offer **discounts** or **free coupons** during the high-order months, and if so, whether this contributes to the increased demand. We can explore the possibility of employing a similar methodology for other months as well, adapting our strategies to optimize order placement throughout the year.

C) During what time of the day, do the Brazilian customers mostly place their orders?

Query -

```
WITH date_and_time_of_day AS (
  SELECT
    order_id,
    order_purchase_timestamp,
    CASE
      WHEN EXTRACT(TIME FROM order_purchase_timestamp) BETWEEN '00:00:00' AND '05:59:59' THEN
'Dawn'
      WHEN EXTRACT(TIME FROM order_purchase_timestamp) BETWEEN '06:00:00' AND '11:59:59' THEN
'Morning'
      WHEN EXTRACT(TIME FROM order_purchase_timestamp) BETWEEN '12:00:00' AND '18:59:59' THEN
'Afternoon'
      ELSE 'Night'
    END AS time_of_day
  FROM `target-brazil-390112.Target_Brazil.orders`
  WHERE EXTRACT(YEAR FROM order_purchase_timestamp) BETWEEN 2016 AND 2018
  GROUP BY order_id, order_purchase_timestamp
)
SELECT time_of_day, COUNT(*) FROM date_and_time_of_day GROUP BY time_of_day ORDER BY COUNT(*) DESC
```

Row	time_of_day	f0_
1	Afternoon	44130
2	Night	28331
3	Morning	22240
4	Dawn	4740

Time of the day

0-6 hrs : Dawn

7-12 hrs : Mornings

13-18 hrs : Afternoon

19-23 hrs : Night

Insights

After analyzing the table, we can say that Brazilian customers tend to place most of their orders in the **Afternoon**. On the other hand, they are less likely to place orders during the early morning hours (**Dawn**).

Recommendation

Through analysis of the table, it has been observed that customers tend to place the majority of their orders during the afternoon hours, indicating a higher likelihood of order placement during this time. Conversely, there is a lower level of activity in terms of order placement during the dawn hours. This could be attributed to reasons such as customers being occupied with work or **unavailable** during those early hours.

To increase order placements during the morning hours, it would be beneficial to study user behaviors, particularly the types of products they typically purchase in the afternoon. By offering limited-time promotions or discounts on those specific products during the morning hours, it is possible to **incentivize** customers and encourage them to place orders during this time. Some examples of such offers could include "**Morning Rush Sale**," "**Early Hours Special**," or "**First Come, First Offer**" service. These initiatives have the potential to boost order placements during the morning periods.

Q.3 Evolution of E-commerce orders in the Brazil region

A) Get the month on month no. of orders placed in each state.

Query -

```
select c.customer_state,
extract(Month from o.order_purchase_timestamp) as month,
count(o.order_id) as total_orders
from `target-brazil-390112.Target_Brazil.orders` as o
join `target-brazil-390112.Target_Brazil.customers` as c
on o.customer_id = c.customer_id
where extract (year from o.order_purchase_timestamp) between 2016 and 2018
group by c.customer_state, month
order by c.customer_state, month
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	month	total_orders
1	AC	1	
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6
11	AC	11	5
12	AC	12	5
13	AL	1	20

Open sort menu

B) How are the customers distributed across all the states?

Query - SELECT

```
  c.customer_state,
  COUNT(DISTINCT c.customer_id) AS number_of_customers
FROM Target_Brazil.orders AS o
JOIN Target_Brazil.customers AS c
  ON o.customer_id = c.customer_id
WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2016 AND 2018
GROUP BY c.customer_state
ORDER BY COUNT(DISTINCT c.customer_id) DESC
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAIL

Row	customer_state	number_of_customer
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020
11	PE	1652

Insights

The highest number of orders is received for **SP- 41746** State and the lowest number of orders is received for **RR-46**

Recommendations

Considering the high number of customers in states such as SP, RJ, and MG, Target could benefit from focusing on these regions. These states exhibit promising market expansion opportunities due to the existing customer demand. Among them, SP stands out with the highest customer count across all states. By prioritizing efforts in SP, Target has the potential to secure a significant **market share** and establish a prominent presence in the retail sector.

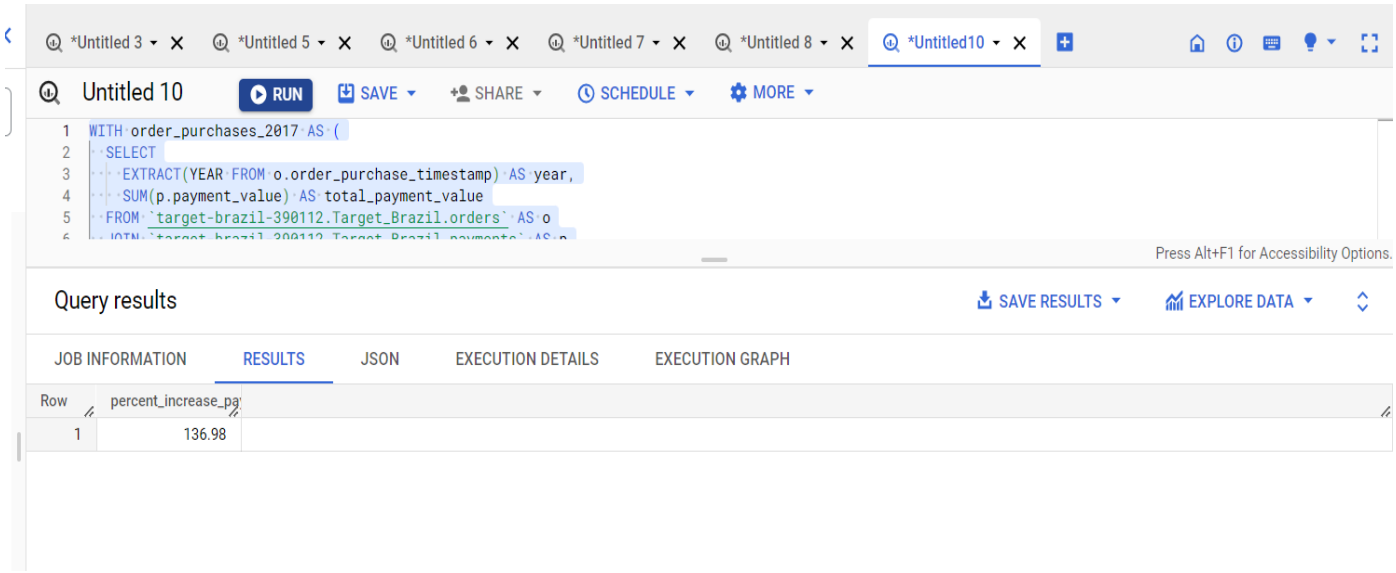
Q.4 Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others

A) Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only). You can use the "payment_value" column in the payments table to get the cost of orders.

Query -

```
WITH order_purchases_2017 AS (
  SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
    SUM(p.payment_value) AS total_payment_value
  FROM `target-brazil-390112.Target_Brazil.orders` AS o
  JOIN `target-brazil-390112.Target_Brazil.payments` AS p
    ON o.order_id = p.order_id
  WHERE EXTRACT(DATE FROM o.order_purchase_timestamp) BETWEEN '2017-01-01' AND '2017-08-31'
```

```
  GROUP BY EXTRACT(YEAR FROM o.order_purchase_timestamp)
), order_purchases_2018 AS (
  SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
    SUM(p.payment_value) AS total_payment_value
  FROM `target-brazil-390112.Target_Brazil.orders` AS o
  JOIN `target-brazil-390112.Target_Brazil.payments` AS p
    ON o.order_id = p.order_id
  WHERE EXTRACT(DATE FROM o.order_purchase_timestamp) BETWEEN '2018-01-01' AND '2018-08-31'
  GROUP BY EXTRACT(YEAR FROM o.order_purchase_timestamp)
)
SELECT ROUND(((b.total_payment_value - a.total_payment_value) / a.total_payment_value) * 100, 2) AS
percent_increase_payment_values
FROM order_purchases_2017 AS a, order_purchases_2018 AS b
```



Insights

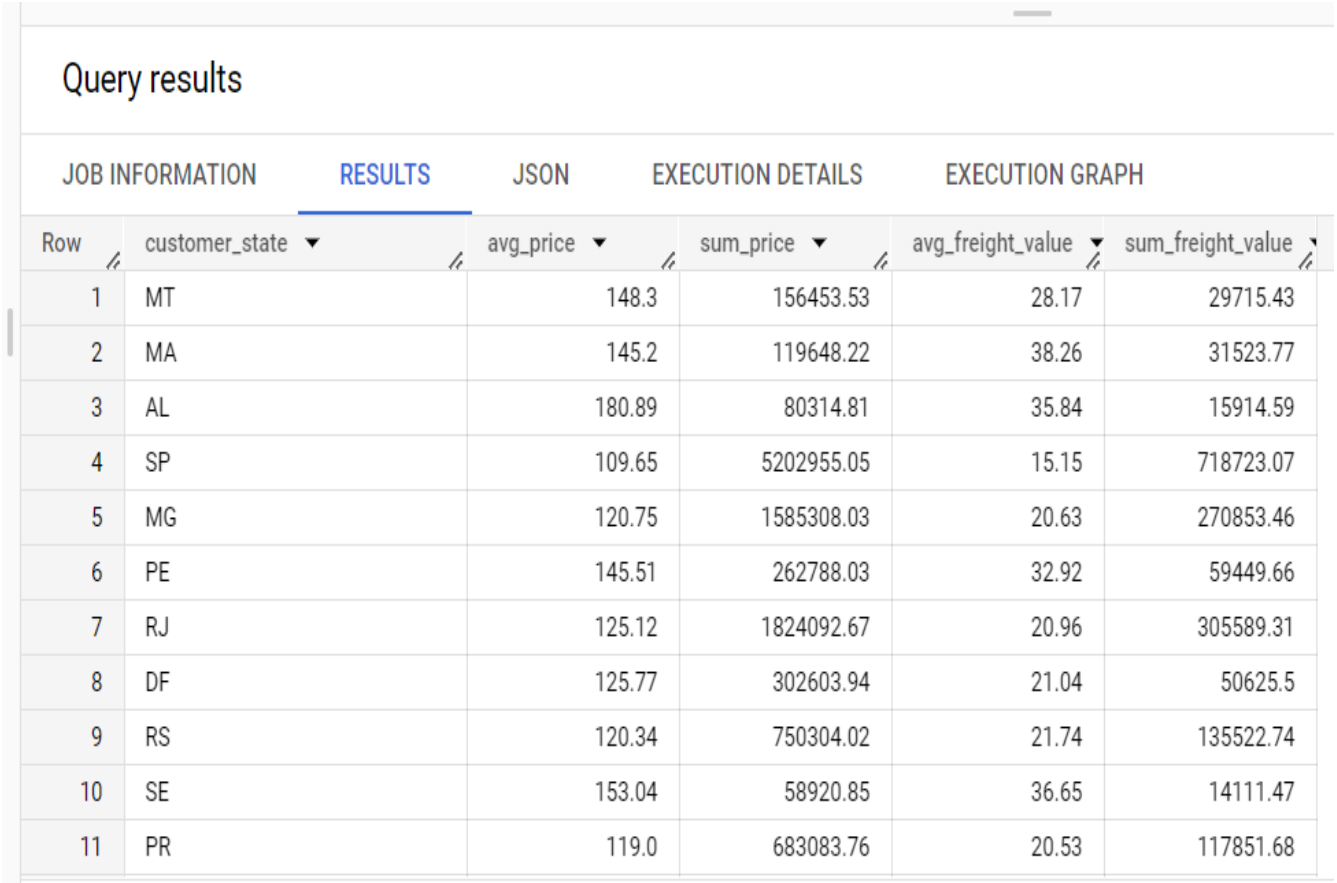
From the above table we can conclude that the increase in the cost of orders from 2017 to 2018 is **137%**

B) Calculate the Total & Average value of order price for each state.

C) Calculate the Total & Average value of order freight for each state.

Query -

```
SELECT
 c.customer_state,
 ROUND(AVG(oi.price), 2) AS avg_price,
 ROUND(SUM(oi.price), 2) AS sum_price,
 ROUND(AVG(oi.freight_value), 2) AS avg_freight_value,
 ROUND(SUM(oi.freight_value), 2) AS sum_freight_value
FROM `target-brazil-390112.Target_Brazil.customers` AS c
JOIN `target-brazil-390112.Target_Brazil.orders` AS o
ON c.customer_id = o.customer_id
JOIN `target-brazil-390112.Target_Brazil.order_items` AS oi
ON o.order_id = oi.order_id
WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2016 AND 2018
GROUP BY c.customer_state
```



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	customer_state	avg_price	sum_price	avg_freight_value	sum_freight_value
1	MT	148.3	156453.53	28.17	29715.43
2	MA	145.2	119648.22	38.26	31523.77
3	AL	180.89	80314.81	35.84	15914.59
4	SP	109.65	5202955.05	15.15	718723.07
5	MG	120.75	1585308.03	20.63	270853.46
6	PE	145.51	262788.03	32.92	59449.66
7	RJ	125.12	1824092.67	20.96	305589.31
8	DF	125.77	302603.94	21.04	50625.5
9	RS	120.34	750304.02	21.74	135522.74
10	SE	153.04	58920.85	36.65	14111.47
11	PR	119.0	683083.76	20.53	117851.68

Q.5 Analysis based on sales, freight and delivery time

A) Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order.

Query -

```
SELECT DATE_DIFF( order_estimated_delivery_date,
order_purchase_timestamp, DAY) AS estimated_time,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY) AS actual_time
FROM Target_Brazil.orders
WHERE DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY) IS NOT NULL
```

Row	estimated_time	actual_time
1	17	30
2	59	30
3	52	35
4	32	30
5	33	32
6	31	29
7	39	43
8	36	40
9	35	37
10	28	33
11	32	38
12	33	36

Query -

```
SELECT DATE_DIFF(order_purchase_timestamp,
order_delivered_customer_date, DAY) AS time_to_delivery,
DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY) AS diff_estimated_delivery
FROM Target_Brazil.orders
```

Row	time_to_delivery	diff_estimated_delive
1	-30	-12
2	-30	28
3	-35	16
4	-30	1
5	-32	0
6	-29	1
7	-43	-4
8	-40	-4
9	-37	-1
10	-33	-5
11	-38	-6
12	-36	-2
13	-34	0

B) Find out the top 5 states with the highest & lowest average freight value

Query – Highest

```
SELECT c.customer_state,
ROUND(AVG(i.freight_value)) AS mean_freight_value,
FROM Target_Brazil.orders o JOIN Target_Brazil.customers c
ON o.customer_id = c.customer_id
JOIN Target_Brazil.order_items i
ON o.order_id = i.order_id
GROUP BY c.customer_state
ORDER BY mean_freight_value DESC
LIMIT 5
```

Row	customer_state	mean_freight_value
1	PB	43.0
2	RR	43.0
3	RO	41.0
4	AC	40.0
5	PI	39.0

Query - Lowest

```
SELECT c.customer_state,
ROUND(AVG(i.freight_value)) AS mean_freight_value,
FROM Target_Brazil.orders o JOIN Target_Brazil.customers c
ON o.customer_id = c.customer_id
JOIN Target_Brazil.order_items i
ON o.order_id = i.order_id
GROUP BY c.customer_state
ORDER BY mean_freight_value asc
LIMIT 5
```

Row	customer_state	mean_freight_value
1	SP	15.0
2	PR	21.0
3	RJ	21.0
4	DF	21.0
5	MG	21.0

C) Find out the top 5 states with the highest & lowest average delivery time

Query -

```
SELECT c.customer_state,
ROUND(AVG(DATE_DIFF(order_purchase_timestamp,
order_delivered_customer_date, DAY))) AS time_to_delivery,
FROM Target_Brazil.orders o JOIN Target_Brazil.customers c
ON o.customer_id = c.customer_id
JOIN Target_Brazil.order_items i
ON o.order_id = i.order_id
```

```
GROUP BY c.customer_state
ORDER BY time_to_delivery DESC
limit 5
```

JOB INFORMATION | RESULTS | JSON | EXECUTI

Row	customer_state	time_to_delivery
1	SP	-8.0
2	PR	-11.0
3	MG	-12.0
4	DF	-13.0
5	RS	-15.0

Query – Lowest

```
SELECT c.customer_state,
ROUND(AVG(DATE_DIFF(order_purchase_timestamp,
order_delivered_customer_date, DAY))) AS time_to_delivery,
FROM Target_Brazil.orders o JOIN Target_Brazil.customers c
ON o.customer_id = c.customer_id
JOIN Target_Brazil.order_items i
ON o.order_id = i.order_id
GROUP BY c.customer_state
ORDER BY time_to_delivery asc
limit 5
```

Row	customer_state	time_to_delivery
1	AP	-28.0
2	RR	-28.0
3	AM	-26.0
4	AL	-24.0
5	PA	-23.0

D) Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

Query - Fastest Delivery

```
SELECT
 c.customer_state,
 ROUND(AVG(oi.freight_value), 2) AS mean_freight_value,
```

```
 ROUND(AVG(DATE_DIFF(order_purchase_timestamp,
order_delivered_customer_date, DAY))) AS mean_time_to_delivery,
 ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY))) AS mean_diff_estimated_delivery
FROM `Target_Brazil.customers` AS c
JOIN `Target_Brazil.orders` AS o
ON c.customer_id = o.customer_id
JOIN `Target_Brazil.order_items` AS oi
ON o.order_id = oi.order_id
WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2016 AND 2018
GROUP BY c.customer_state
ORDER BY AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY)) DESC
LIMIT 5
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	customer_state	mean_freight_value	mean_time_to_delive	mean_diff_estimated
1	AC	40.07	-20.0	20.0
2	RO	41.07	-19.0	19.0
3	AM	33.21	-26.0	19.0
4	AP	34.01	-28.0	17.0
5	RR	42.98	-28.0	17.0

Query – Slowest Delivery

```
SELECT
 c.customer_state,
 ROUND(AVG(oi.freight_value), 2) AS mean_freight_value,
 ROUND(AVG(DATE_DIFF(order_purchase_timestamp,
order_delivered_customer_date, DAY))) AS mean_time_to_delivery,
 ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY))) AS mean_diff_estimated_delivery
FROM `Target_Brazil.customers` AS c
JOIN `Target_Brazil.orders` AS o
ON c.customer_id = o.customer_id
JOIN `Target_Brazil.order_items` AS oi
```

```
ON o.order_id = oi.order_id
WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2016 AND 2018
GROUP BY c.customer_state
ORDER BY AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY))
LIMIT 5
```

Row	customer_state	mean_freight_value	mean_time_to_delive	mean_diff_estimated
1	AL	35.84	-24.0	8.0
2	MA	38.26	-21.0	9.0
3	SE	36.65	-21.0	9.0
4	ES	22.06	-15.0	10.0
5	BA	26.36	-19.0	10.0

Q.6 Analysis based on the payments

A) Find the month on month no. of orders placed using different payment types

Query -
```
SELECT DISTINCT p.payment_type,
EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
COUNT(p.order_id) OVER (PARTITION BY p.payment_type
ORDER BY EXTRACT(YEAR FROM order_purchase_timestamp) ASC,
EXTRACT(MONTH FROM order_purchase_timestamp) ASC) AS total_orders
FROM Target_Brazil.orders o JOIN Target_Brazil.payments p
ON o.order_id = p.order_id
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	payment_type	year	month	total_orders
1	UPI	2016	10	63
2	UPI	2017	1	260
3	UPI	2017	2	658
4	UPI	2017	3	1248
5	UPI	2017	4	1744
6	UPI	2017	5	2516
7	UPI	2017	6	3223
8	UPI	2017	7	4068
9	UPI	2017	8	5006
10	UPI	2017	9	5909
11	UPI	2017	10	6902
12	UPI	2017	11	8411
13	UPI	2017	12	9571

Insights

Credit card & **UPI** are most used method for transaction in Brazil.

B) Find the no. of orders placed on the basis of the payment installments that have been paid

Query -

```
SELECT payment_installments,
COUNT(order_id) AS total_orders
FROM Target_Brazil.payments
GROUP BY payment_installments
```

JOB INFORMATION | RESULTS | JSON

Row	payment_installment	total_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644
11	10	5328

Insights & Recommendations

It would be beneficial to offer discounts and/or lower EMIs for customers opting for higher numbers of installments. This approach can potentially incentivize customers to choose larger installment plans and result in increased order volumes.